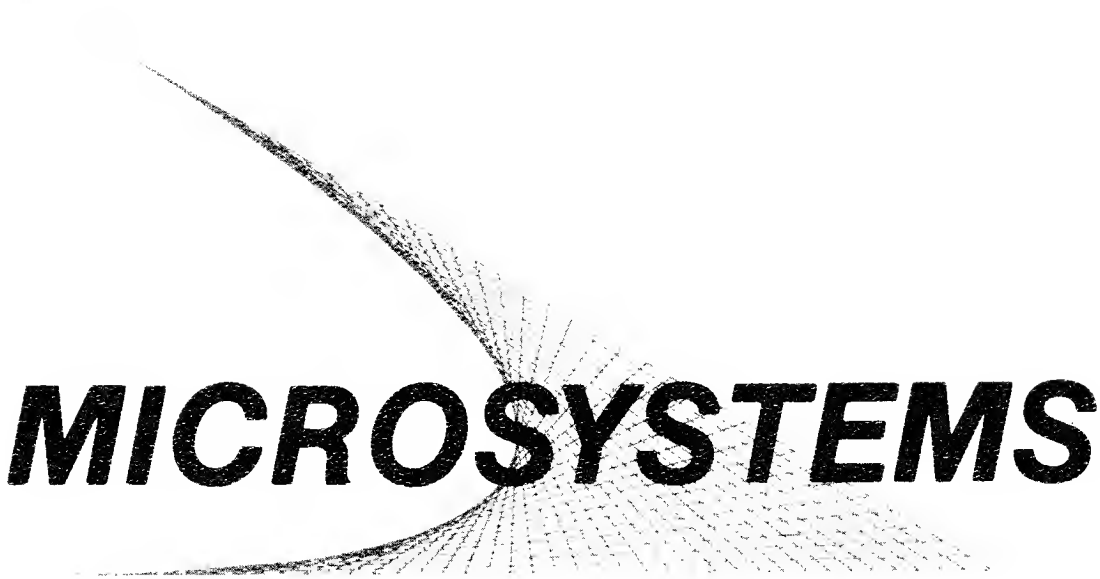
MOTOROLA

M68KEDIT/D9

M68000 Family
CRT Text Editor
User's Manual



QUALITY • PEOPLE • PERFORMANCE

MOTOROLA

M68KEDIT/D9

NOVEMBER 1985

M68000 FAMILY

CRT TEXT EDITOR

USER'S MANUAL

Ninth Edition

Eighth Edition April 1984

MICROSYSTEMS

REVISION RECORD

M68KEDIT/D9 -- October 1985. Replaces M68KEDIT/A1 and M68KEDIT/D8, modifies syntax conventions (boldface type), introduces consecutive page numbering, makes minor corrections to the text, and adds a keyword index.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

1

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

The function of an editor is to enter, modify, or delete text in ASCII files. Motorola's disk-based editor offers a versatile and easy-to-use means to accomplish these tasks.

Three levels of control -- page, command, and insert -- are provided for source program entry and editing. Two of these, page and command, are used in the CRT mode of operation, and two levels, insert and command, are used in the line mode of operation (refer to Table 1-1). Users having EXORterm 155-equipped systems and VME/10 systems normally edit in the CRT mode. Users of other terminals must edit under the line mode of operation. However, when editing under direction of an executing chainfile, only the command level of the line mode is permitted; the insert level is not permitted.

TABLE 1-1. Levels of Control

CRT MODE	LINE MODE
page level	command level
command level	insert level

NOTE

Users of non-EXORterm 155 or VME/10 terminals may prefer to use the VERSAdos Terminal-Independent Editor (TIE), which allows editing in the CRT mode under VERSAdos 4.5 and subsequent. Refer to the VERSAdos Terminal-Independent Editor User's Manual.

Source program files produced through use of the editor are in indexed sequential ASCII or sequential ASCII form and can be edited, as can the ASCII listing files produced by the various VERSAdos assemblers and compilers. Files in sequential format are placed in a temporary scratch file for editing, and saved only when the **QUIT** command is executed. However, indexed sequential files are edited directly -- a much faster method. Future editing time can often be saved by choosing the indexed sequential format and by reformatting old files to this format.

Keyed indexed sequential files with fixed-length records are not properly handled by the editor.

1

NOTE

Indexed files created with this release of the editor cannot be edited using versions of the editor supplied with VERSAdos operating system diskette labeled SYS0 System SW 1.00, 5/9/80 and prior.

To reformat an old indexed file for editing with this release, invoke the editor (refer to Chapter 2), specifying the old filename in the input field, a new filename in the output field, and the I option in the options field on the command line. At least one change must be made for the old file to be reformatted and saved.

1.2 OPERATING ENVIRONMENT

The editor executes under VERSAdos on the VME/10 Microcomputer System, the EXORmacs Development System, and a system based on any of the following: VERSAmodules 03 and 04 or VMEmodules MVME101, MVME110, MVME117, MVME120/121, MVME122/123, or MVME130/131.

1.3 CONFIGURING EXORterm 155 FOR EDITOR OPERATION

M68000 Family systems incorporate a great deal of capability for accommodating the various interface, memory, communications, and other elements with which a microcomputer system is implemented. Much of the communications flexibility is included in EXORterm 155, which provides access to individual options by means of three groups of rear panel switches. For an EXORterm directly connected to the system, when using the editor, these switches should be set as shown in Figure 1-1.

1.4 COMMAND SYNTAX: FORMAT AND RULES

1.4.1 Format

Syntactic descriptions in this manual use a modified Backus-Naur Form (BNF). A brief description of its symbols and their meanings are:

boldface string	A boldface string is a literal such as a command or a program name, and is to be typed just as it appears.
< >	Angle brackets enclose a "syntactic variable", that is replaced in a command line by one of a class of items it represents.
|	This symbol indicates that a choice is to be made. One of several items separated by this symbol should be selected.
[ ]	Square brackets enclose an item that is optional. The enclosed item may occur zero or one time.
[ ]...	Square brackets followed by periods enclose an item that is optional/repetitive. It may appear zero or more times.

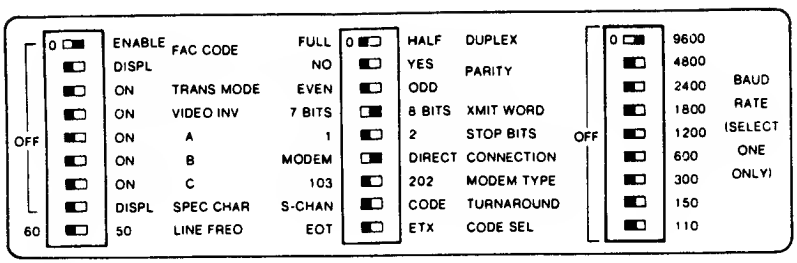


FIGURE 1-1. EXORterm 155 Rear Panel Switches

1.4.2 Rules

A space must separate a command mnemonic from any parameter field that follows it.

All command line mnemonics and arguments, when entered at the keyboard or from a chainfile, may be in uppercase or lowercase, with one exception: A string parameter in a **FIND** or **CHANGE** command must match the file data exactly -- in uppercase and lowercase (e.g., new and New do not match).

Each editor command and data line is terminated by a carriage return (pressing the **RETURN** or ↵ key). This is shown in examples as (**CR**), only when necessary for clarity; e.g., when a carriage return is the only input required.

1

1.5 RELATED DOCUMENTATION

The following publications may provide additional helpful information. If not shipped with this product, they may be obtained from Motorola's Literature Distribution Center, 616 West 24th Street, Tempe, AZ 85282; telephone (602) 994-6561.

DOCUMENT TITLE	MOTOROLA PUBLICATION NUMBER
System Generation Facility User's Manual	M68KSYSGEN
M68000 Family VERSAdos System Facilities Reference Manual	M68KVSF
VERSAdos Messages Reference Manual	M68KVMSG
VERSAdos Data Management Services and Program Loader User's Manual	RMS68KIO
VERSAdos Terminal-Independent Editor User's Manual	M68KTIE

CHAPTER 2

INVOKING THE EDITOR

2.1 INVOCATION

The editor is invoked from VERSAdos by entering a command line after the system prompt (=). Figure 2-1 summarizes the means of entry into the editor operating modes and levels, as described in Chapters 3 and 4.

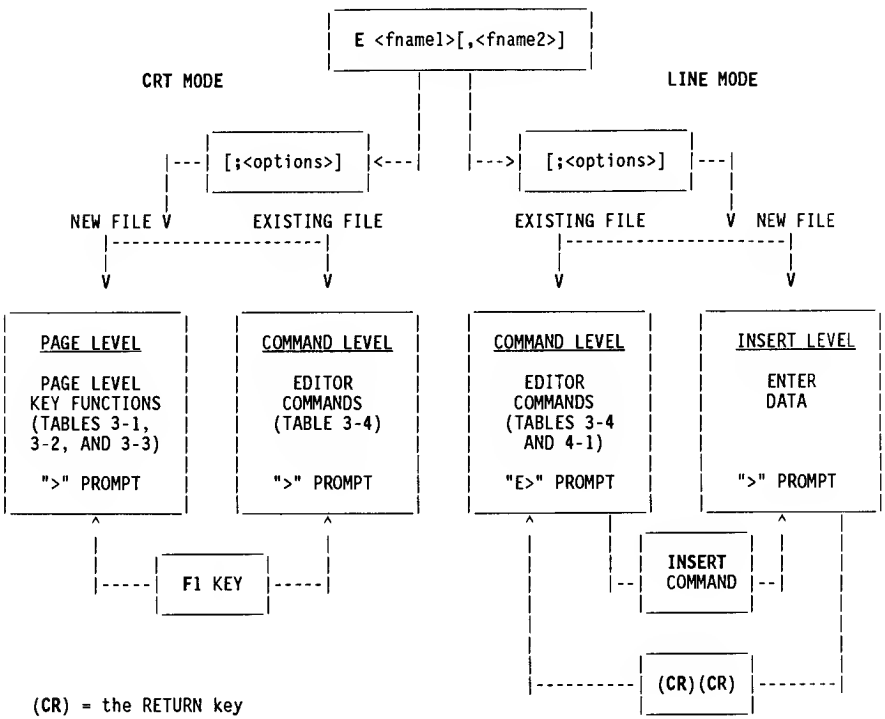


FIGURE 2-1. Editor Mode/Level Entry

The format of the command is:

=E <fname1>[,<fname2>][;<options>]

where:

<fname1> is the VERSAdos file descriptor (explained in the VERSAdos System Facilities Reference Manual) of an ASCII file. The format is:

<volume name>:<user number>.<catalog>.<filename>.<ext>

It describes either an old (existing) file or a new (non-existing) file. If existing, the file contents are made available for editing. If <fname2> is not specified, the contents of the edit file are stored under <fname1> on completion (upon issuing a **SAVE** or **QUIT** command with no arguments). If non-existing, <fname1> is newly created to receive the contents of the edit file on completion. A default value of .SA is assumed for <ext>, the extension.

<fname2> is the VERSAdos file descriptor (refer to <fname1>) of a new (non-existing) file. If <fname2> is specified, <fname1> must be an existing file. <fname2> receives the contents of the edit file upon completion, and the contents of <fname1> are left unchanged. A default value of .SA is assumed for <ext>, the extension.

<options> is one or more of the following options (multiple options are entered with no intervening spaces):

L Enter the line mode of operation (refer to Chapter 3). All the other options, as described below, are valid for use in the line mode. The default condition is the CRT mode.

Line mode is the normal condition for chainfile editing; i.e., the **L** option need not be specified on the command line within a chainfile.

NOTE

Parameters 6 and/or R of the CRTDCB macro in the IOC.<driver>.AG file can be changed for each terminal on the system to indicate terminal type. The default is an EXORterm 155. If this is changed to be a non-EXORterm 155 terminal and the system is reSYSGENed, the editor goes directly into line mode without the user having to specify the **L** option. The same result can be obtained on-line with the **CONFIG** utility by changing item 16 in the Terminal Parameters menu. Refer to the description of the CONFIG utility in the M68000 Family VERSAdos Facilities Reference Manual, for instructions on changing the parameters.

K This option allows the viewing of a file without truncation of records greater in length than 79 characters. When the editor is called and the K option is not specified, records are truncated to 79 characters each. Viewing of the first 79 characters in each record is provided. Although no changes to a file can be made, the editor scroll functions and the **DOWN, FIND, LIST, QUIT, RANGE, UP,** and **VERIFY** commands are available. The **K** option may only be used alone or in combination with the **L** option. The **PRINT** command may be used with the **K** option.

I Create the output file (<fname2> or new <fname1>) in indexed sequential format. Refer to Note (1).

S Create the output file (<fname2> or new <fname1>) in sequential format. Refer to Note (1).

A Sets assembler tabulation stops (tabs) (columns 1, 11, 18, 37). Refer to Note (2).

C Sets COBOL tabs (columns 1, 6, 9, 12). Refer to Note (2).

F Sets FORTRAN tabs (column 1, 7). Refer to Note (2).

P Sets Pascal tabs (columns 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 73). Refer to Note (2).

NOTES

(1) The **S** and **I** options are valid for a new <fname1> or new <fname2> descriptor only, and are ignored otherwise. The following table shows the output file type, based on the command line option and the input file type (ID = indexed sequential, S = sequential):

OPTIONS	INPUT FILE TYPE (NEW)	ID	S
(NONE)	ID	ID	S
I	ID	ID	ID
S	S	S	S

(2) When no tab option is specified, a default format of tab stops every ten columns is used.

2.2 EXAMPLES

=E OLD

The existing file under the default <volume>:<user number>.<catalog>. OLD.SA is opened and input to the edit file. In CRT mode the first 20 records of the file would be displayed on the screen. On termination of the edit (entering a **QUIT** command), the file OLD.SA receives the newly edited records.

=E OLD,NEW;IA

The input file is OLD.SA and the output filename is NEW.SA. The file type is indexed sequential, and assembler tabs are set in columns 1, 11, 18, and 37.

=NEW;F

A new file is opened under the name NEW.SA, which receives the contents of the edit file upon termination. It is an indexed sequential file (the default type for a new file) with FORTRAN tabs set in column 1 and 7.

=E LISTING.LS;K

The existing file LISTING.LS is opened; the file may then be viewed, but no changes may be made. This option is helpful when searching for character strings for reference only, and a printout is not desired -- for example, when searching for errors in an assembly listing. With sequential files -- such as an assembly listing -- time is saved also because, when using the **K** option, a scratch file is not first created to hold the file contents for editing.

CHAPTER 3

THE CRT MODE OF OPERATION

3.1 INTRODUCTION

In the CRT mode, users can edit while operating at the page level and at the command level. Unless the **L** option is specified on the VERSAdos command line, the CRT mode is automatically entered when the editor is called. The **L** option changes the default to line mode. The line mode of editing is also the default if the user specifies a non-EXORterm 155 in the configuration for that port, or if the user has an EXORterm 150.

When creating a new file, the editor enters the page level. If called for the purpose of editing an existing file, it enters command level. Either level can be reached from the other through use of the **F1** function key.

A window is formed on the screen within which the new file is created or a portion of the existing file is placed for editing. The window occupies the top 20 lines of the screen. Below the window, line 21 is reserved for display of messages and the name of the file being edited. Line 22 contains the command pointer (>); when the blinking cursor is moved to this line (by pressing the **F1** key), editor commands may be entered here. Lines 23 and 24 are used to identify the special Fn function keys located on the top row of the keyboard (refer to Table 3-1 or Table 3-2). Also on line 23 is the word LINE, below which is the number of the current line (line pointed to by the record pointer (>)). Refer to Table 3-3 for cursor commands.

Each record of a file occupies an individual line on the screen. (The terms record and line are used synonymously in this manual. They refer to a line in the file that may either be blank or contain text.)

3.2 PAGE LEVEL EDITING

In the page level, editing is achieved through combined use of the usual typewriter-equivalent keys, the cursor movement keys, and use of the special function keys on an EXORterm (refer to Table 3-1) or a VME/10 (refer to Table 3-2).

When the editor is called to edit an existing file, the first 20 lines (records) of the file appear in the screen window with the record pointer (>) positioned at the first line of text. The cursor resides on line 22 with the command pointer. The record pointer may be moved up or down within the window, but it always remains in column 0. By pressing the **F1** function key, the cursor is moved to line 1 (or the same line as the record pointer). The cursor remains on the same line as the record pointer, but it can then be moved anywhere within the 79-column range. The six cursor keys (↑ , ↓ , →, ←, ⇥, ⇤) move both the pointer and/or cursor. Both the record pointer and cursor wrap around from the top and bottom of the window; the cursor wraps around from the left and right sides of the window. Changes in text are made by positioning the cursor over the character to be changed and pressing function and/or alphanumeric keys. Both pages and lines may be scrolled on the screen through the use of the special function keys **F3**, **F4**, **F5**, and **F6**.

3.2.1 Exiting the Editor from Page Level

From page level, the user returns to command level by pressing the **F1** function key. This positions the cursor after the command pointer on line 22. The **QUIT** command (refer to paragraph 3.16) can then be entered to save the edit file, exit the editor, and return to VERSAdos. If no changes to the file have been made during the edit session, no output file is made.

3

3.2.2 Example

The following example shows the editor being called and used in page level to make a correction to a source program segment (filename TEST.SA). The editor is called by the command line:

=E **TEST** (User entry is in boldface type)

Momentarily, the following two lines are displayed on the screen:

```
VERSAdos EDITOR RELEASE x.xx
Copyright xxxx by Motorola Inc.
```

Then the following screen image is displayed:

```
+---------------------------------------------------------------------------+
|>START EQU  *                                                              |
|       MOVE.L  D1, SAVSESS     SAVE SESSION NUMBER                         |
|       MOVE.L  D2, SAVVOLN     SAVE VOLUME NAME                            |
|       MOVE.L  A0, SAVENAME    SAVE TASK NAME                              |
|       LEA.L   STACK,A7        SET UP STACK POINTER                        |
|       BSR     GETCMD          GO DO READ                                  |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                    SYS :2560..TEST   .SA  |
|>                                                                          |
| F1    LINE     F3      F4       F5      F6      F7      F11      F12      |
|MODE    1      PAGE^   PAGEv    LINE^   LINEv   DUP     SET TAB  DEL TAB   |
|                                                                           |
+---------------------------------------------------------------------------+
```

Four or five steps are required to change line 5 and produce the screen image below:

a. Press the **F1** key to enter page level.

b. Press the cursor-down key four times (the cursor moves to the fifth line).

c. Using the cursor-right key, position the cursor over the period in LEA.L.

d. Press the space bar twice (.L is replaced by two spaces).

OR

d. Press the **DEL CHAR** (EXORterm) or the **DCHR** (VME/10) key twice to delete .L (the text to the right of .L moves left two places as two character spaces are deleted).

e. Press the **INS CHAR** (EXORterm) or the **ICHR** (VME/10) key twice (the text to the right of the cursor moves right two places as two character spaces are inserted).

To save the updated file in TEST.SA and exit the editor, press the **F1** key to enter the command mode, then enter the **QUIT** command. The message EDIT DONE is displayed and the VERSAdos prompt (=) is given.

```
+-------------------------------------------------------------------------+
|                                                                         |
| START  EQU *                                                            |
|        MOVE.L   D1, SAVSESS      SAVE SESSION NUMBER                    |
|        MOVE.L   D2, SAVVOLN      SAVE VOLUME NAME                       |
|        MOVE.L   A0, SAVENAME     SAVE TASK NAME                         |
|>       LEA      STACK,A7         SET UP STACK POINTER                   |
|        BSR      GETCMD           GO DO READ                             |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                 SYS :2560..TEST    .SA  |
| >                                                                       |
|  F1    LINE       F3      F4       F5      F6      F7       F11      F12 |
| MODE    5        PAGE^   PAGEv    LINE^   LINEv   DUP    SET TAB  DEL TAB|
|                                                                         |
|                                                                         |
+-------------------------------------------------------------------------+
```

TABLE 3-I. Page Level Key Functions -- EXORterm Only

KEY LABEL	SHIFTED/ UNSHIFTED	SCREEN ACTION
CLEAR HOME	S/U	Sends the cursor to the home position (line I, column I). The screen contents are unaffected.
DEL	S	Deletes the character to the left of the cursor.
_	U	Underscore. Overwrites existing character.
INS CHAR	S/U	The character in the cursor column and all characters to the right of the cursor are moved right one column. The character in column 79 is lost. The cursor does not change position, and a blank is inserted at the cursor position.
DEL CHAR	S/U	Deletes the character in the cursor column. The cursor does not change position. All columns to the right of the cursor are moved left one column. Column 79 receives a blank character.
INS LINE	S/U	All lines below and including the cursor line are moved down one line. The cursor maintains its position in the line.
DEL LINE	S/U	Deletes the line at the current cursor position.
LINE ERAS	S	Replaces all 79 characters in the cursor line with blanks. The cursor moves to column I in that line.
PAGE	U	Not supported.
BREAK	S/U	Exits the current operation, goes to command level.
RETURN	S/U	Carriage return. In command level, sends a command for execution.
F1	U	From page level, enters command level. From command level, enters page level.
F2		Not used.
F3	U	Scrolls forward one page.
F4	U	Scrolls backward one page.
F5	U	Scrolls forward one line.
F6	U	Scrolls backward one line.

TABLE 3-1. Page Level Key Functions -- EXORterm Only (cont'd)

KEY LABEL	SHIFTED/ UNSHIFTED	SCREEN ACTION
F7	U	Duplicates the character immediately above the cursor.
F8-F10		Not used.
F11	U	Sets a tab stop in the column where the cursor resides. A maximum of 20 tab stops are held by the editor.
F12	U	Deletes the tab stop in the column where the cursor resides.

TABLE 3-2. Page Level Key Functions -- VME/10 Only

KEY LABEL	SHIFTED/ UNSHIFTED	SCREEN ACTION
↖	S/U	Sends the cursor to the home position (line 1, column 1). The screen contents are unaffected.
DEL	S	Deletes the character to the left of the cursor.
1 ICHR	U	The character in the cursor column and all characters to the right of the cursor are moved right one column. The character does not change position, and a blank is inserted at the character position.
A DCHR	U	Deletes the character in the cursor column. The cursor does not change position. All characters to the right of the cursor are moved left one column. Column 79 receives a blank character.
2 ILINE	U	All lines below and including the cursor line are moved down one line. The cursor maintains its position in the line.
B DLINE	U	Deletes a line at the current cursor position.
EOL	S/U	Replaces all 79 characters in the cursor line with blanks. The cursor moves to column 1 in that line.

TABLE 3-2. Page Level Key Functions -- VME/10 Only (cont'd)

KEY LABEL	SHIFTED/ UNSHIFTED	SCREEN ACTION
BREAK	S/U	Exits the current operation, goes to command level.
↵	S/U	Carriage return. In command level, sends command for execution.
F1	U	From page level, enters command level. From command level, enters page level.
F2		Not used.
F3	U	Scrolls forward one page.
F4	U	Scrolls backward one page.
F5	U	Scrolls forward one line.
F6	U	Scrolls backward one line.
F7	U	Duplicates the character immediately above the cursor.
F9-F10		Not used.
F11	U	Sets a tab stop in the column where the cursor resides. A maximum of 20 tab stops are held by the editor.
F12	U	Deletes the tab stop in the column where the cursor resides.
F13-F16		Not used.

TABLE 3-3. Page Level Key Functions -- VME/10 and EXORterm

KEY LABEL	SHIFTED/ UNSHIFTED	SCREEN ACTION
←	S/U	Moves the cursor one position to the left. From column 1, the cursor wraps around to column 79.
→	S/U	Moves the cursor one position to the right. From column 79, the cursor wraps around to column 1.
↑	S/U	Moves the cursor up one line. From line 1, the cursor wraps around to line 20.
↓	S/U	Moves the cursor down one line. From line 20, the cursor wraps around to line 1.
⇤	S/U	Moves the cursor left to the next tab stop. At end-of-line, it stops in column 1; if pressed again, it wraps around to the right-most tab stop.
⇥	S/U	Moves the cursor right to the next tab stop. After the right-most tab stop, it wraps around to column 1.

3

3.3 COMMAND LEVEL EDITING

When the command level is entered, the cursor is positioned to the right of the command pointer (>) on line 22 of the CRT screen. Any of the commands listed in Table 3-4 and described in paragraphs 3.4 through 3.21 may be entered at this point.

The record pointer (>), in the data window of the screen, is always positioned at the current record (line) in the edit file. The pointer is moved by editor commands, by key with up or down arrows, or by use of the **F3**, **F4**, **F5**, and **F6** keys.

NOTE

The following commands are also valid when using the editor in line mode. In the command level of line mode, the prompt is preceded by the letter E (E>).

TABLE 3-4. Editor Commands -- CRT and Line Mode

COMMANDS	DESCRIPTION
ADUP	Copy records from the file and place them in the XTRACT buffer, appending them to any records already in the buffer. The records still remain in the file.
AMOVE, AMOV	Move records from the file and place them in the XTRACT buffer, appending them to any records already in the buffer. The records moved are deleted from the file.
CHANGE, C	Change strings within records.
DELETE, DELE, DEL	Delete records or strings from the edit file.
DOWN, D	Move the record pointer downwards.
DUPLICATE, DUP	Copy records and place them a newly created XTRACT buffer. The records still remain in the file.
EXTEND, EX	Append data to the end of records.
FIND, F	Find a string, or position record pointer at a record.
LINE	Display the line number of the current record.
MERGE	Merge specified records from another VERSAdos file into the edit file.
MOVE	Move records into a newly created XTRACT buffer. The records moved are deleted from the file.
PRINT	Output records to the printer.
QUIT, Q	Save the edit file in a VERSAdos file, terminate the edit session, and return control to VERSAdos.
RANGE,R	Establish default values for the vertical and horizontal ranges of the **CHANGE**, **FIND**, **PRINT**, and **SAVE** commands.
SAVE	Save part or all of the edit file in another VERSAdos file, and continue editing.
TAB	Specify tab stops.
UP, U	Move the record pointer upwards.
XTRACT, X	Copy the records from the XTRACT buffer and insert them above the current record in the file. The records remain in the buffer.

3.3.1 Vertical and Horizontal Ranges

Many of the editor commands allow the specification of a vertical and/or horizontal range. Default values, however, are assumed whenever a range is not specified. For the **CHANGE**, **FIND**, **PRINT**, and **SAVE** commands, the default ranges may be changed by the **RANGE** command.

The vertical range format is:

a. [<startrec>[-<endrec]] -- where <startrec> is the record number at which to begin; <endrec> is the record number at which to stop, or in a **FIND**, **MERGE**, **PRINT**, or **SAVE** command, <endrec> defaults to the last record in the file. In other commands, the starting record is also the ending record.

b. [*[-<num>]] -- where * is the current record and <num> is the number of records (including the current record) on which to perform the function. <num> defaults to a value of 1.

NOTE

A period (.) can be used in place of the asterisk (*) in all commands that specify vertical range.

The horizontal range format is:

a. [:<startcol>[-<endcol>]] - where <startcol> is the column on each line at which to begin; <endcol> is the column on each line at which to stop. Column number corresponds to character number reckoned from the start of the line (record). End of line assumed when <endcol> is not specified.

b. [:F<i>[-F<i+n>]] - where <i> is the i'th tab stop at which to begin and <i+n> is the tab stop at which to stop. The range is inclusive of the columns from F<i> to F<i+n>. When using tab stops to specify the horizontal range, the "F" must be input along with the tab stop number; e.g., **[:F1-F9]**.

3.3.2 Example

The four or five steps described in the example for page level editing (refer to paragraph 3.2.2) could be replaced by one command:

```
C 5/LEA.L/LEA  /
```

To keep the other columns from shifting to the left, it is necessary to add two spaces to replace .L.

3.3.3 ADUP

The **ADUP** command duplicates one or more records from the file and places them in the XTRACT buffer, appending them to any records already existing there. The records remain in the file as well as the buffer. The format of the command is:

ADUP [<vert>]

where:

<vert> specifies the vertical range of the records to be duplicated. Its format is described in paragraph 3.3.1. The default vertical range is the current record.

If no XTRACT buffer exists, the buffer created is ten times the number of records being duplicated, where each record is assumed to be 79 characters in length.

The buffered records are re-inserted in the edit file by means of an **XTRACT** command, and the buffer is cleared by an **XTRACT** A command (refer to paragraph 3.21).

If the buffer is filled up at any time, a BUFFER FULL message is displayed. The buffer contents should be moved to the file and the buffer cleared. Any records beyond the XTRACT buffer size are not saved.

The position of the record pointer is not changed.

EXAMPLES:

>ADUP Copy the current record into the XTRACT buffer. If an XTRACT buffer exists, the record is appended to the records already in the buffer. If no XTRACT buffer exists, one is created.

>ADUP 1-10 Copy records 1 through 10 into the XTRACT buffer. Append them to the end of the XTRACT buffer if one exists; otherwise, create a new XTRACT buffer for the records.

3.3.4 AMOVE

AMOVE
AMOV

The **AMOVE** command moves one or more records from the file and places them in the XTRACT buffer, appending them to any records already existing there. The records are deleted from the file. The format of the command is:

AMOVE [<vert>]
AMOV [<vert>]

where:

<vert> Specifies the vertical range of the records to be moved. Its format is described in paragraph 3.3.1. The default vertical range is the current record.

If no XTRACT buffer exists, the buffer created is ten times the number of records being moved, where each record is assumed to be 79 characters in length.

The buffered records are re-inserted in the edit file by means of an **XTRACT** command, and the buffer is cleared by an **XTRACT A** command (refer to paragraph 3.21).

If the buffer is filled up at any time, a BUFFER FULL message is displayed. The buffer contents should then be moved to the file and the buffer cleared.

On completion, the record pointer is positioned at the record following the last moved (deleted) record.

EXAMPLES:

>AMOV Move the current record to the XTRACT buffer, appending to the end of the records in the buffer if one exists; otherwise, create a new buffer for the record. Delete the record from the file.

>AMOV 1-10 Move records 1 through 10 into the XTRACT buffer, appending if the XTRACT buffer exists. Delete these records from the file.

>AMOV *-9999 Move the current record and all succeeding records (9999 usually ensures all records to the end of file) to the XTRACT buffer, appending if the XTRACT buffer exists. Delete these records from the file.

3.3.5 CHANGE

The **CHANGE** command searches a portion of or an entire file for a string and changes it to a desired string. The format of the command is:

CHANGE [<vert>][:<horiz>][,<trans>][;<occur>][<dc><str1><dc><str2><dc>][<cnt>]
C [<vert>][:<horiz>],[<trans>][;<occur>][<dc><str1><dc><str2><dc>][<cnt>]

where:

<vert> is the vertical range of the records to be changed. Its format is described in paragraph 3.3.1. The default range is the current line (unless changed by the **RANGE** command; refer to paragraph 3.17).

<horiz> is the horizontal range within each record that may be searched and changed. Its format is described in paragraph 3.3.1. The default range is the entire record (unless changed by the **RANGE** command).

<trans> is a transparent character that may appear one or more times in <str1>. When searching, the editor looks for the non-transparent character(s) of <str1>, ignoring what appears in the file in the position(s) corresponding to the transparent character. However, the length of the string deleted and replaced by <str2> includes the transparent character position(s). The transparent character has no significance in/on the content of <str2>.

<occur> is either <num> or the letter **A**, where:

- <num> is a decimal value. It specifies which occurrence of <str1> in each record is to be changed to <str2> -- for example, if <num> is 3, the third occurrence of <str1> in each record of the vertical range is changed to <str2>.
- **A** requests that all occurrences of <str1> in the horizontal range be changed to <str2>.

The default value for <occur> is 1.

<dc> is a delimiting character of the user's choice which encloses the string fields. The character must be non-blank and non-numeric, and cannot be a comma (,) or a semicolon (;).

<str1> is the target string of characters.

<str2> is the character string to which the target, <str1>, is to be changed.

<cnt> is a count specified as either <num> or the letter **A**, where:

<num> is a decimal value. It specifies the number of records in the vertical range in which <str1> is to be changed to <str2>.

A requests that the change occur in the entire vertical range specified.

The default value for <cnt> is 1.

When a **CHANGE** command is specified with no parameters, the **CHANGE** command last specified is executed, starting at the current record position. When <str1> and a null <str2> are specified, <str1> is deleted. When <str2> and a null <str1> are specified, <str2> is inserted at the beginning of the horizontal range.

Following execution of the **CHANGE** command, the record pointer is positioned at the last affected record.

EXAMPLES:

>C /SAM/BILL/	Change the first occurrence of SAM in the current line to BILL.
>C 100-200/SAM/BILL/	Change the first occurrence of SAM to BILL. Search each line from 100 through 200.
>C 100-200/SAM/BILL/A	Change the first occurrence of SAM to BILL in all lines from 100 through 200.
>C 100-200;2/SAM/BILL/A	Change the second occurrence of SAM to BILL in each line from 100 through 200.
>C ;2/SAM/BILL/	Change the second occurrence of SAM to BILL on the current line.
>C 120/BOB//	Delete BOB from line 120.
>C //BOB/	Insert BOB at beginning of current line.
>C *-5/SAM/BILL/ or **>C .-5/SAM/BILL/**	Search the next five lines, and change the first occurrence of SAM to BILL.
>C	Repeat the last **CHANGE** command entered, starting at the current record. In this example sequence, change the next occurrence of the string SAM to BILL.

>C 100:30-60/SAM/BILL/	On line 100 and in columns 30 through 60 only, change the first occurrence of SAM to BILL.
>C 100-500,#/FA##2/FA2/A	Change the first occurrence of the 5-character string FA##2 (where # means that any character can be in that position in the file) to FA2 on all lines from 100 through 500.
>C 1-200:F2-F3/SAM/BILL/15	On lines 1 through 200 and between tab stops 2 and 3 inclusive, change the first occurrence of SAM to BILL in the first 15 lines where SAM is found.
>C 1-9999;A/SAM/BILL/A	Change every occurrence of SAM to BILL in the file.

3.3.6 DELETE

DELETE
DELE
DEL

The **DELETE** command removes one or more records or portions of records from the edit file. The format of the command is:

DELETE [<vert>][:<horiz>]
DELE [<vert>][:<horiz>]
DEL [<vert>][:<horiz>]

where:

<vert> is the vertical range within each record that may be deleted. Its format is described in paragraph 3.3.1. The default range is the current line.

<horiz> is the horizontal range within each record that may be deleted. Its format is described in paragraph 3.3.1. The default range is the entire record.

Specifying a vertical range allows a group of consecutive records to be deleted from the file. Specifying a horizontal range causes the data within that range to be deleted from each record within the vertical range. If desired, the entire file could be deleted by entering a vertical range larger than the largest record number in the file.

When deleting an entire record, the record following the last record affected is the new position of the record pointer. When only deleting within a horizontal range, the last record affected is the new position of the record pointer.

EXAMPLES:

>DEL	Delete the entire current line.
>DEL 10-100	Delete lines 10 through 100.
>DEL *-20:10-50 or **>DEL .-20:10-50**	Delete all data within columns 10 through 50 on the next 20 lines.
>DEL :15-30	Delete all data within columns 15 through 30 of the current line.
>DEL 1-999999:1-1	Delete all data in column 1 of the entire file.
>DEL 1-999999:1-5	Delete all data within columns 1 through 5 of the entire file.

DELETE
DELE
DEL

>DEL 1-999999	Delete the entire file. This does not work if the number is greater then eight (99,999,999) digits.
>DEL 1-10:2	Delete all data within columns 2 through 79 on lines 1 through 10.

3.3.7 DOWN

DOWN
D

The **DOWN** command moves the record pointer downwards a specified number of records. The format of the command is:

DOWN [<num>]
D [<num>]

where:

<num> is the number of records to advance past. The default value is 1.

If the pointer is moved past the end-of-file, the pointer stops at the last record in the file (in line mode, the message BOF OR EOF ENCOUNTERED is displayed).

EXAMPLES:

>D 15 Move the record pointer down 15 lines.

>DOWN 100 Move the record pointer down 100 lines.

>D Move the record pointer down 1 line.

3.3.8 DUPLICATE

DUPLICATE
DUP

The **DUPLICATE** command copies one or more records from the file and places them in the XTRACT buffer, leaving the file contents unchanged. The format of the command is:

DUPLICATE [<vert>]
DUP [<vert>]

3

where:

<vert> is the vertical range of the records to be copied. Its format is described in paragraph 3.3.1. The default range is the current record.

Each use of the command deletes the old XTRACT buffer and creates a new one. Any data existing in the buffer, prior to executing **DUP**, is lost. The new buffer size in characters is 79 times the number of records duplicated. For example, if 10 records are duplicated, a 790 character buffer is created. (Extra room in the buffer allows for possible use of **ADUP** or **AMOV** commands to append to the buffer.)

Records are re-inserted in the file by means of an **XTRACT** command, and the buffer may then be cleared by an **XTRACT A** command (refer to paragraph 3.21).

The position of the record pointer is not changed.

EXAMPLES:

>DUP	Copy the current record into the XTRACT buffer.
>DUP 100-200	Copy all records from 100 through 200 into the XTRACT buffer.
>DUP *-5 or **>DUP .-5**	Copy five lines, beginning with the current line, into the XTRACT buffer.

3.3.9 EXTEND

The **EXTEND** command appends data to the end of a record or group of consecutive records in the file. The format of the command is:

EXTEND [<vert>]<dc><string><dc>
EX [<vert>]<dc><string><dc>

where:

<vert> is the vertical range of the records to be extended. Its format is described in paragraph 3.3.1. The default range is the current record.

<dc> is a delimiting character which encloses the string. It may be any non-numeric, non-blank character.

<string> is the string characters to be appended to the end of each record in the vertical range. Characters that extend beyond column 79 are lost.

The record pointer is positioned at the beginning of the last record affected.

EXAMPLES:

>EX /./ Append a period to the end of the current record. (Slashes are convenient delimiting characters.)

>EX *-10/;/ Append a semicolon to the end of the next 10 records.
or
>EX .-10/;/

>EX /SAM/ Append the string SAM to the end of the current record.

3.3.10 FIND

FIND
F

The **FIND** command can either locate a character string within the file, or position the record pointer at a specific record. The format of the command is:

FIND [<vert>][:<horiz>][,<trans>][;<occur>][<dc><string><dc>][<count>]
F [<vert>][:<horiz>][,<trans>][;<occur>][<dc><string><dc>][<count>]

3

where:

<vert> is the vertical range of the <string> or record to be found. Its format is described in paragraph 3.3.1. The default range is the entire file (unless changed by the **RANGE** command; refer to paragraph 3.17).

<horiz> is the horizontal range within each record wherein the search for the <string> takes place. Its format is described in paragraph 3.3.1. The default range is the entire record (unless changed by the **RANGE** command).

<trans> is a transparent character that may appear one or more times in <string>. When searching, the editor looks for the non-transparent character(s) of <string>, ignoring what appears in the file in the position(s) corresponding to the transparent character.

<occur> is either <num> or the letter **A**, where:

<num> is a decimal value. It specifies which occurrence of <string> in a record is to be found - for example, if <num> is 2, the records where <string> occurs a second time will be found and displayed.

A is the same as the default, where <occur> has a value of 1.

The default value for <occur> is 1.

<dc> is a delimiting character. It may be any character that is not a blank, an alphanumeric, a comma (,), or a semicolon (;).

<count> is either <num> or the letter A, where:

<num> is a decimal value. It specifies the number of records in which <string> is to be found and displayed.

A requests that all records containing <string> are to be found and displayed.

The default value for <count> is 1.

When a **FIND** command is executed with no parameters, it implies a request to re-execute the last **FIND** command, starting at the current record plus one.

When the command is executed with only the vertical range parameter of <startrec> (refer to paragraph 3.3.1) specified, it implies a request to position the record pointer at the specified record. **FIND 0** implies a request to position the record pointer at the beginning of the file and, conversely, **FIND 999999** (or a number greater than that of the final record) implies a request to position the record pointer at the end of the file.

If the screen becomes full while displaying records, the display scrolls up until the list is exhausted. The display may be paused at any time by pressing the **CTRL** and **S** keys, and resumed by pressing any key. The display may be aborted by pressing the **BREAK** key, followed by entering any editor command.

At the end of the **FIND** display, the screen can be returned to its normal display by pressing the **RETURN** key or entering any editor command.

EXAMPLES:

>F 0	Move the record pointer to the beginning of the file.
>F 140	Move the record pointer to line 140 in the file.
>F 999999	Move the record pointer to the end of the file.

NOTE

Any number greater than the number of the last record in the file can be used.

>F /SAN/	Move the record pointer to the line containing the first occurrence of the string SAM.
>F	Re-execute last entered **FIND** command line starting from the current record plus one. In this example sequence, it would find the next occurrence of SAM.
>F SAM/A	Find all occurrences of the specified string in the file starting at the beginning of the file. Display only lines containing the string. Scroll upward if screen capacity is exceeded.

3.3.11 LINE

The **LINE** command provides the line number (record number) corresponding to the current record. The format of the command is:

LINE

On execution, the message CURRENT **LINE** IS x is displayed, where x is the current record number. In CRT mode, the message is displayed directly above the command line.

Line numbers are not part of the file, but are used by the editor to communicate with the user.

EXAMPLE:

>LINE

```
CURRENT LINE IS    1
>LINE
```

Message appears directly above the command line in CRT mode.

3.3.12 MERGE

The **MERGE** command inputs all or a group of consecutive records from another VERSAdos file and inserts them into the edit file above the current record. The format of the command line is:

MERGE [<vert>] <file>

where:

<vert> is the vertical range of the records in the specified VERSAdos file. Its format is described in paragraph 3.3.1. The default range is the entire VERSAdos file.

<file> is the descriptor of a VERSAdos ASCII file. The file descriptor format is:

<volume name>:<user number>.<catalog>.<filename>.<extension>

Only the <filename> portion is required in the command argument. However, if a <user number> is specified, then a <volume name> must also be specified. <extension> has a .SA default value.

The data from the VERSAdos file is inserted into the edit file immediately before the current record (at least one record must be in the edit file, or an error message is generated).

The position of the record pointer is not changed. It continues pointing to the pre-merge record.

EXAMPLES:

>MERGE SAM Insert the entire file SAM.SA into the edit file above the current record.

>MERGE 68-89 VOL3:313..OLD

Insert records 68 through 89 of the VOL3:313..OLD.SA file into the edit file above the current record.

3.3.13 MOVE

The **MOVE** command moves one or more records from the file to the XTRACT buffer, then deletes the record(s) from the file. The format of the command is:

MOVE [<vert>]

where:

<vert> is the vertical range of the records to be moved. Its format is described in paragraph 3.3.1. The default range is the current record.

Each use of the command deletes the old XTRACT buffer and creates a new one. Any data existing in the buffer prior to executing **MOVE** is lost. The new buffer size in character is 79 times the number of records moved. For example, if 10 records are moved, a 790 character buffer is created. (Extra room in the buffer allows for the possible use of the **ADUP** and **AMOVE** commands to append to the buffer.)

Records are reinserted in the file by means of an **XTRACT** command, and the buffer may then be cleared by an **XTRACT A** command (refer to paragraph 3.21).

the record pointer is positioned at the record following the last moved (deleted) record.

EXAMPLES:

>MOVE Move the current record to the XTRACT buffer and delete that record from the file.

>MOVE *-5
or
>MOVE .-5 Move the next 5 records to the XTRACT buffer and delete them from the file.

3.3.14 PRINT

The **PRINT** command attaches the line printer and prints all or a portion of the file. The format of the command line is:

PRINT [<vert>] [<device] [<option>]

where:

<vert> is the vertical range of the records to be printed. Its format is described in paragraph 3.3.1. The default is the entire file (unless changed by the **RANGE** command).

<device> is the VERSAdos device name of the line printer; e.g., #PR1. The default device name is #PR.

<option> may be one of the letters **D** or **T**, where:

D indicates double spacing -- that is, one blank record is printed between each record.

T indicates triple spacing -- that is, two blank records are printed between each record.

The default is single spacing -- that is, records are printed exactly as they occur in the edit file.

Three blank records are printed before printing of the actual edit file records begin.

If the Spool system task (SPL) is running, it creates an output file which is then printed according to the SPL sequencing rules.

The command does not alter the screen display.

EXAMPLES:

>PRINT Print the entire file, single-spaced, on device #PR.

>PRINT *-100 D Print the next hundred records, double-spaced, on device #PR.
or
>PRINT .-100 D

>PRINT 5-8 T Print records 5 through 8, triple-spaced, on device #PR.

>PRINT #PR1 Print the entire file, single-spaced, on device #PR1.

>PRINT 55 #PR2 Print records 55 to the end-of-file, single spaced, on device #PR2.

>PRINT *-20 Print the next twenty records on device #PR.
or
>PRINT .-20

3.3.15 QUIT

The **QUIT** command will save the edit file in a VERSAdos file, exit the editor, and return to the VERSAdos level. The format of the command is:

```
QUIT [A]
Q [A]
```

where:

A is an option to abort the editing operation, deleting the edit file and leaving the original contents of <fname1> intact (refer to Chapter 2). This option is valid only when <fname1> and/or <fname2> is a sequential file; otherwise, it is ignored.

The edit file is closed in its present state and transferred to a VERSAdos file, as described in Chapter 2. If the **A** option is specified on the command line for a sequential file, the original unedited contents of the file will be saved.

When the editor is called to edit a sequential file, it creates a scratch file in which the contents of the file or new data are placed for editing. Should an I/O error occur on any Close, whether the Close was initiated by quitting the editor or by a system error, the <user number>, <catalog>, and <filename> under which the edit file was to be saved (refer to Chapter 2) is given to the scratch file. The file is then assigned and written to the <volume name> on which <fname1> exists or is to be stored. The contrived extension .EA is added to the <filename>, because the identical <filename> with a standard <extension> might already exist, preventing completion of the assignment. Extension .EB or .EC, etc., would be used if <fname1>.EA already exists from previously saving a scratch edit file.

Following such an occurrence, the VERSAdos **RENAME** utility should be used to give the saved scratch file the desired name and extension.

EXAMPLES:

>QUIT End editing session and save edit file; return to VERSAdos command level.

>QUIT A Abort editing session on sequential file: the edit file is lost, while the original contents of <fname1> are left intact. Return to the VERSAdos command level.

>Q End the editing session and save the edit file; return to VERSAdos.

3.3.16 RANGE

RANGE
R

The **RANGE** command changes the default vertical and horizontal ranges of the **FIND**, **CHANGE**, **PRINT**, and **SAVE** commands. The original default ranges are in the descriptions of these commands. The format of the command is:

RANGE [<vert>][:<horiz>]
R [<vert>][:<horiz>]

where:

<vert> is the new vertical range. Its format is described in paragraph 3.3.1.

<horiz> is the new horizontal range. Its format is described in paragraph 3.3.1.

When neither vertical nor horizontal ranges are specified on the command line, default values previously established through use of the **RANGE** command are deleted. Defaults return to the original default ranges for each command.

EXAMPLES:

>R 1-100 Set lines 1 through 100 as the vertical range for the **FIND**, **CHANGE**, **PRINT**, and **SAVE** commands.

>R :10-30 Set columns 10 through 30 as the default horizontal range for the **FIND**, **CHANGE**, **PRINT**, and **SAVE** commands.

>R 1-1--:10-30 Set lines 1 through 100 as the default vertical range and columns 10 through 30 as the default horizontal range for the **FIND**, **CHANGE**, **PRINT** and **SAVE** commands.

>RANGE Delete any ranges previously established by the **RANGE** command; return to original default ranges.

3.3.17 SAVE

The **SAVE** command stores all or specified portions of the edit file in a new VERSAdos file. The format of the command is:

SAVE [<vert>] [<file>]

where:

<vert> is the vertical range of the records to be saved. Its format is described in paragraph 3.3.1. The default range is the entire file (unless changed by the **RANGE** command).

<file> is the descriptor for a non-existing VERSAdos file. The file type is indexed sequential. The file descriptor format is:

<volume name>:<user number>.<catalog>.<filename>.<extension>

Only the <filename> portion is required in the command argument. <extension> has a .SA default value.

The user's default <volume name>;<user number>, and <catalog> can be used.

This command is useful when editing to ensure minimal data loss in case of system failure. Execution of the command does not alter the screen display.

EXAMPLES:

>SAVE NEW Save the entire edit file in a new file named NEW.SA.

>SAVE *-100 NEW1
or
>SAVE .-100 NEW1 Save the next 100 lines of the edit file in a new file named **NEW1**.SA.

>SAVE Write the entire edit file contents to disk. Continue editing in the same mode and level. The default file descriptor is taken from the <fname2> specified on the command line to VERSAdos that called the editor (refer to Chapter 2). If <fname2> was not specified, the edit file is written to <fname1>.

3.3.18 TAB

The **TAB** command sets or changes the tabulation stops. The format of the command is:

TAB [<num>[,<num>]...|<letter>]

where:

<num> is a column position at which a tab stop is to be set. Multiple stops, separated by commas, may be specified. Stops are added to those already set by default, a command line option, or a previous **TAB** command, and to column 1.

<letter> is one of the letters:

A Clear all tabs, then set assembler tabs in columns 1, 11, 18, and 37.

C Clear all tabs, then set COBOL tabs in columns 1, 6, 9, and 12.

F Clear all tabs, then set FORTRAN tabs in columns 1 and 7.

P Clear all tabs, then set Pascal tabs in columns 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, and 73.

Either a sequence of numbers (<num>) or a letter (<letter>), but not both, may be a parameter on the command line.

Entered tab values are sorted and saved in ascending order. A maximum of 20 tab stops are held by the editor at a time.

By executing a **TAB** command line with no parameters, tabs can be reset to the default format of every 10 columns, or to the tabs corresponding to the option specified on the editor VERSAdos command line.

Execution of the **TAB** command does not alter the screen display.

In CRT mode, with EXORterms only, the forward tab key may be transferred to another key by pressing the **SET TAB** key, then the key for another character. That key then becomes the forward tab key. The function can be returned to the original key by reversing the procedure.

EXAMPLES:

>TAB Tabs revert to the editor call option settings or, if none were specified, to the default format of every 10 columns.

>TAB 20,50 Add tabs in columns 20 and 50.

TAB

>TAB 30	Add a tab in column 30.
>TAB A	Clear all tabs, then set assembler tabs in columns 1, 11, 18, and 37.
>TAB P	Clear all tabs, then set Pascal tabs in columns 1, 4, 7, etc.

3.3.19 UP

UP
U

The **UP** command moves the record pointer upwards a specified number of records. The format of the command is:

UP [<num>]
U [<num>]

where:

<num> is the number of records above which to move. The default value is 1.

Execution of the **UP** command causes the record pointer to be moved upwards the specified number of records. If the pointer is moved past the beginning-of-file, the pointer stops at the first record in the file (also, in line mode, the message BOF OR EOF ENCOUNTERED is displayed).

EXAMPLES:

>UP 50	Move the record pointer up 50 records.
>U 5	Move the record pointer up 5 records.
>U	Move the record pointer up 1 record.

3.3.20 XTRACT

XTRACT
X

The **XTRACT** command transfers the contents of the XTRACT buffer to the file. It is used in conjunction with the **MOVE**, **DUPLICATE**, **AMOVE**, and **ADUP** commands for relocating records in the edit file. The format of the command is:

```
>XTRACT [A]
>X [A]
```


where:

A is an option to clear the XTRACT buffer. The records in the buffer are not transferred into the file, but are lost.

Instead of a record number specified on the command line, the record pointer must be positioned where the records are to be inserted. The **XTRACT** command copies the records from the XTRACT buffer to the edit file immediately above the current record. The record remains positioned at the pre-merge record.

The records transferred into the file still remain in the XTRACT buffer. They are only cleared from the buffer by execution of the command with the **A** option -- that is, **XTRACT A**.

EXAMPLES:

>X Copy the records contained in the XTRACT buffer into the edit file above the current record.

>X A Clear the XTRACT buffer, leaving the edit file and screen display unchanged.

CHAPTER 4

THE LINE MODE OF OPERATION

4.1 INTRODUCTION

The normal mode of operation of the VME/10 or EXORmacs when operated from its own keyboard, is the CRT mode, described in Chapter 3. The line mode of operation described in this chapter would be appropriate for use with an auxiliary terminal such as Motorola's EXORterm 150, or in chain mode operation.

The line mode is brought up by typing **;L** on the command line that calls the editor.

The insert level is entered automatically when editing a file which contains no records. This level is specified by the prompt >. In this level, any characters entered at the keyboard are placed in the edit file. The characters are stored as records, where a record is a line of characters that is terminated by the carriage return key. The insert level, however, is not allowed when executing from a chainfile. (For more information on the insert level, refer to paragraph 4.2.3, **INSERT**.) To enter or return to the command level, enter a carriage return alone at the insert level prompt.

The command level is entered automatically when editing a file which contains records. This level is specified by the prompt E>. In this mode, any of the 24 CRT and line mode commands (refer to Tables 3-4 and 4-1) may be entered. The 18 commands usable in CRT mode and line mode are described in Chapter 3. The six commands whose usage is restricted to line mode are described in this chapter. To return to insert level, enter an **I** command.

When deleting the entire contents of a file in line mode, the mode is automatically switched to insert level.

An edit session in line mode is terminated by typing **Q** or **QUIT** to the command level prompt, E>.

EXAMPLE:

```
=E EDITFILE;L     (call editor in line mode)
>  (insert level; enter records)
        .
        .
        .
>(CR)  (go to command level)
E>LIST
 .
 .
 .
E>Q  (return to VERSAdos)
=
```

Note that in the examples shown in Chapter 3, the CRT mode editor prompt (>) is shown. In line mode, the commands must be entered to the line mode/command level prompt (E>), as shown in the examples in this chapter.

4.2 LINE MODE COMMANDS

All of the commands described in Chapter 3 can be used in line mode. In addition, line mode has six of its own commands -- **COLM**, **DTAB**, **INSERT**, **LIST**, **STAB**, and **VERIFY** -- which are summarized in Table 4-1 and described in the following paragraphs.

TABLE 4-1. Editor Commands -- Line Mode Only

COMMANDS	DESCRIPTION
COLM	Display the ruler of column spacings.
DTAB	Delete tab stops.
INSERT, I	Enter insert level, for adding records to the edit file.
LIST, L	List records in the edit file.
STAB	Specify tab stops.
VERIFY, V	Display records that are altered or record pointer changes.

4.2.1 COLM

The **COLM** command is used to display a ruler of the column spacing across a record. Its purpose is informational only -- to count the columns -- and can be specified at any time while in command level. The format of the command is:

COLM

On execution, a ruler is displayed as follows:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
```

where dashes (-) specify single columns, a plus (+) specifies a five-column interval, and a number (1, 2, etc.) specifies a ten-column interval.

NOTE

The text displayed by a **LIST** command is left-shifted one position from the column ruler.

EXAMPLE:

```
E>COLM

----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
E>
```

4.2.2 DTAB

The **DTAB** command is used to delete tab stops set by a command line option, or a **TAB** or **STAB** command. The format of the command is:

DTAB

On execution, the ruler of column spacings is displayed, followed on the next line by a question mark, as follows:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
?
```

where dashes (-) specify single columns, a plus (+) specifies a five-column interval, and a number (1, 2, etc.) specifies a ten-column interval.

On the same line as the ?, press the space bar to move to the column where the stop is to be deleted. In that column, enter an asterisk (*). If additional stops are to be deleted, space again and enter asterisks in the columns where those stops exist. The line is then terminated by pressing the carriage return key.

NOTE

The text displayed by a **LIST** command is left-shifted one position from the column ruler.

EXAMPLE:

```
E>DTAB
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
?     *  *         *(CR)
E>
```

Tab stops are deleted from columns 7, 10, and 20.

4.2.3 INSERT

INSERT
I

The **INSERT** command allows the user to enter records into the file. Records are entered above the current record. The format of the command is:

INSERT [<offset>]
I [<offset>]

where:

<offset> is either <num> or the letter **A**, where:

<num> is the decimal value. It specifies the number of records past the current record at which to begin inserting data.

A requests that records to be inserted be appended at the end of the file.

On execution, the command level prompt E> is replaced by the insert level prompt >. Records can then be entered into the file.

Each record is terminated by pressing the carriage return key. If two returns are entered consecutively, the prompt E> is printed and control is returned to the command level. (To insert a blank record, press the space bar at least once after the prompt >, before pressing the return key.)

The insert level is entered automatically when there is an empty file (namely, when creating a new file or editing an empty existing file, or when a **DELETE**, **MOVE**, or **AMOVE** clears the file).

NOTE

The **INSERT** command is not allowed when executing from a chainfile.

A typical display of prompts is seen in this example of inserting two lines of assembly language code:

```
E>I
>START    RA      PROC
>PROC     MOVE.L  #$1000,A7
>(CR)
E>
```

EXAMPLES:

E>INSERT Begin inserting data at the present record pointer position -- i.e., the current line.

E>I 5 Begin inserting data five records beyond the current line.

E>I A Append inserted records at the end of the file.

4.2.4 LIST

LIST
L

The **LIST** command lists all or a specified portion of a file on the terminal. The format of the command is:

LIST [<vert>]
L [<vert>]

where:

<vert> is the vertical range of the records to be listed. Its format is described in paragraph 3.3.1. The default range is the entire file.

If the parameter is a single record specification --e.g., **LIST 104** -- it must be a number within the range of the file. Otherwise, the error message xxx IS LAST LINE is displayed, where xxx is the number of the last record in the file.

On execution, the current record pointer is positioned at the first record listed.

EXAMPLES:

E>LIST	List the entire file.
E>L *-100 or **E>L .-100**	Starting from the current record, list 100 records.
E>L 5-8	List records 5 through 8.
E>L 35	List record 35.

4.2.5 STAB

The **STAB** command is used to set tab stops. A maximum of 20 stops may be set. Stops are added to any stops set by default (every 10th column), a command line option, or a **TAB** command. The format of the command is:

STAB

On execution, the ruler of column spacings is displayed, followed on the next line by a question mark, as follows:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
?
```

where dashes (-) specify single columns, a plus (+) specifies a five-column interval, and a number (1, 2, etc.) specifies a ten-column interval.

On the same line as the ?, press the space bar to move to the column where a stop is to be set. In that column, enter an asterisk (*). If additional stops are to be set, space again and enter asterisks in those columns where stops are desired. The line is then terminated by pressing the carriage return key.

When entering text by means of the **INSERT** command, the tab key is pressed to begin typing at the next tab stop. These tabs are entered in the text, but are not displayed on the input line for the EXORterm 150 or 155. On a VME/10, they are displayed as "^I". However, if the text is printed on a printer using a **LIST** command, although shown in column format, it does not reflect the actual locations of the tabulations.

NOTE

Initially, the editor has default tab stops at columns 1, 10, 20, 30, 40, 50, 60, and 70. The **STAB** command merely adds to these. To delete any other stops, use the **DTAB** command.

EXAMPLE (VME/10):

```
E>STAB

----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
?    *       *      *(CR)
>E>I
>^IJOHN^IMARY^ISAM^IGEORGE
>^IA^IB^IC^ID
>^I1^I2^I3^I4
>(CR)
E>LIST

             JOHN MARY SAM GEORGE
              A    B    C    D
              1    2    3    4
E>
```

STAB

Tabs are set in columns 7, 15, 22, and 33 in addition to those already set by default or previous **STAB** command. Note that the display for the **LIST** command does not reflect the actual tab settings.

4.2.6 VERIFY

The **VERIFY** command allows a record to be displayed when a command function has altered the record, or when the current record pointer is newly positioned to that record. The format of the command is:

VERIFY [OFF]
V [OFF]

where:

OFF is an option to disable the verify function.

The current record display that follows execution of an **AMOVE**, **CHANGE**, **DELETE**, **EXTEND**, **FIND**, **MERGE**, **MOVE** or **XTRACT** command is omitted when the verify function is disabled. Other messages that follow execution of these commands, however, are not altered.

The initial and default state of the function is enabled.

EXAMPLES:

E>**V** Enable the verify function.

E>**V OFF** Disable the verify function.

4

THIS PAGE INTENTIONALLY LEFT BLANK.

CHAPTER 5

TEXT EDITOR MESSAGES

5.1 GENERAL

Table 5-1 is a list of the messages that might be encountered when using the text editor. Messages are output directly by the editor and do not use the facilities of the Error Message Handler (EMH).

Some of the messages include the expression DO=. When such a message is output, DO= is followed by eight hexadecimal digits having the format xx0000zz. If xx = 18, then zz is an entry in Table 5-2. If xx = 10, then zz is an entry in Table 5-3. Refer to the VERSAdos Messages Reference Manual for detailed message information.

TABLE 5-1. Text Editor Error Messages

MESSAGE	DESCRIPTION
xxxx IS LAST LINE	The vertical range requested is past the end-of-file.
xxxxxxxx LINES DELETED	Number of lines deleted by **DELETE** or **MOVE** command.
xxxxxxxx RECORDS SAVED - BUFFER FULL	During the execution of an **AMOV** or **ADUP**, the XTRACT buffer cannot hold any more records. The buffer is moved by an **XTRACT** command and cleared by an **XTRACT A** command.
2 LOGICAL UNITS NOT AVAILABLE	Editor exited.
ASQ ERROR DO=	Terminates editor -- unable to allocate an Asynchronous Service Queue (ASQ).
BOF OR EOF ENCOUNTERED	Beginning-of-file or end-of-file encountered. Continue.
BREAK WAS INPUT	**BREAK** key was pressed during page or command execution -- editor goes to command level.
CANNOT HANDLE BREAKS DO=	Editor cannot handle **BREAK** key during this edit session -- continues with edit.

TABLE 5-1. Text Editor Error Messages (cont'd)

MESSAGE	DESCRIPTION
COMMAND ABORTED BY BREAK KEY	**BREAK** key was pressed during PRINT or **FIND** command. Refer to the command descriptions for action.
COMMAND LINE ERROR	Editor command line is incorrect. Re-enter.
CLOSE ERROR DO=	File not closed. Editor exited.
DATA OVERRUN I/O ERROR	Screen I/O error -- editor rewrites current page to screen and goes to command mode.
DELETE ERROR DO=	Save files. Editor exited.
DEVICE NOT READY	Device not ready.
ERROR - CHANGE ACCESS DO=	Internal error -- attempt failed to change access to public read/write for logical unit 5. Editor exited.
ERROR - IOS CALL DO=	Screen input/output error. Files saved. Editor exited.
ERROR - MERGE NOT EXECUTED	System error.
ERROR - RETRIEVE ATTRIBUTES DO=	Internal error -- attempt failed on FHS Retrieve-Attributes-for-Console call. Editor exited.
FILE DOESN'T EXIST	Check descriptor entered on command line.
FILENAME2 EXISTS. OVERWRITE (Y/N)?	Yes -- continues edit with <fname2> as output file. No -- terminates edit session.
ILLEGAL FHS COMMAND CODE x	Internal editor error. Files saved. Editor exited.
ILLEGAL INPUT FILENAME DO=	Editor exited.
ILLEGAL OUTPUT FILENAME DO=	Editor exited.
INCOMPATIBLE FILE TYPES - NO MERGE	File to be merged must be of edit file type.
INSERT ERROR DO=	Save files. Editor exited.

TABLE 5-1. Text Editor Error Messages (cont'd)

MESSAGE	DESCRIPTION
INSUFFICIENT DIRECTORY SPACE	Editor exited.
INSUFFICIENT DISK SPACE	1) On Open: Editor exited. 2) On Close or Save: Output file may not be saved. Input file may not be closed. Editor exited. 3) Otherwise: files saved; editor exited.
INTERNAL LIST ERROR	Editor cannot process command -- re-enter command.
INVALID CHARACTER RECEIVED - REINPUT	During CRT mode editing, the editor found a character it couldn't handle. The screen is rewritten and command level is entered.
INVALID COMMAND-NO FILE RECORDS	No records are in the file for which a **CHANGE**, **FIND**, **MERGE**, etc., is requested. These commands require at least one record.
INVALID KEYSIZE - OUTPUT FILE	Open error -- keysizes of input and output files do not match -- editor exited.
INVALID OPTION WITH NEW FILENAME	Editor exited.
INVALID OPTIONS	Exit editor.
INVALID TAB OPTIONS - USING DEFAULTS	Edit session continues.
I/O FRAMING ERROR	Screen I/O error. Editor rewrites current page to screen and goes to command mode.
LUN NOT AVAILABLE - NOT EXECUTED	**SAVE** or **MERGE** error; no Logical Unit Number (LUN) for file.
MAXIMUM OF TAB SET (20)	20 tab stops have already been set. No more tabs may be set until some are deleted.
NEW FILENAME1-FILENAME2 IGNORED - CONTINUE (Y/N)?	<fname1> is a new (non-existing) file; therefore, <fname2> is not required.
NO DATA MOVED TO XTRACT BUFFER	Try **MOVE** or **DUPLICATE** using fewer records at a time.

TABLE 5-1. Text Editor Error Messages (cont'd)

MESSAGE	DESCRIPTION
NO RECORDS INSERTED - BUFFER FULL	During an **AMOVE** or **ADUP**, the XTRACT buffer cannot hold any more records. Move the buffer records to the file by an **XTRACT** command, then clear the buffer by an **XTRACT A** command.
OPEN ERROR DO=	Editor exited.
OUTPUT FILE EXISTS - SAVE NOT DONE	**SAVE** command requires new output file.
PRINTER BUSY	Reenter command.
PRINTER NOT READY	Check printer readiness. Reenter command.
READ ERROR DO=	Save files. Editor exited.
RECORD NOT ON FILE	1) On read: save files; exit editor. 2) On delete or insert: message displayed. Continue.
REPLACE ERROR DO=	Save files. Editor exited.
SAVE ERROR DO=	Output file not saved. Input file may not be closed. Editor exited.
SAVE ERROR - FILE NOT CREATED	System error.
SEARCH FOR STRING ERROR DO=	Internal error. Save files. Editor exited.
STRING NOT FOUND	String not found during **FIND** or **CHANGE** command execution.
SYNTAX ERROR-EDIT TERMINATED	An invalid command was entered in chain mode.
USING K OPTION	A command was attempted that is invalid while editing with the **K** option.
WHAT?	Syntax error -- re-enter command.
XTRACT BUFFER DELETED	Buffer memory de-allocated.
XTRACT BUFFER DOESN'T EXIST	XTRACT buffer is created by executing **MOVE**, **DUPLICATE**, **AMOVE**, or **ADUP** command only.

5.2 VERSAdos FHS/IOS ERROR CODES

The VERSAdos philosophy of employing logical I/O, rather than hardware I/O, allows all of the activity to be carried out by a single Input/Output Services (IOS) module. Another module, FHS, performs all File Handling Services. On completion of the activity required by any call to either of these service modules, a status byte is returned to the using entity. The byte contains zero if the call was carried out successfully. Otherwise, it contains a number keyed to the general type of error encountered.

Error codes are divided into two major categories -- FHS errors and IOS errors. For FHS errors, the high order bit of the status byte is cleared; for IOS errors, the bit is set. In addition, IOS errors are further categorized as follows:

TYPE OF IOS ERROR	STATUS BYTE BITS SET	
Parameter block	7	(8x)
Device independent	7, 6	(Cx)
Device dependent	7, 6, 5	(Ex)
Channel	7, 6, 5, 4	(Fx)

It should be noted that an IOS error status can be returned on an FHS call since the call may require disk I/O.

A general description of all FHS and IOS error codes is provided in the following tables. Table 5-2 lists FHS error codes and their meanings, and Table 5-3 provides IOS error codes and meanings. Greater insight into a specific error occurrence than is provided by the tables may require reference to other M68000 family system manuals (such as the VERSAdos Messages Reference Manual and the VERSAdos Data Management Services and Program Loader User's Manual).

TABLE 5-2. File Handling Services (FHS) Error Messages

zz	DESCRIPTION
00	No error.
01	FHS trap server does not exist. Probably caused by system problem.
02	Invalid command. Command specified is not valid for device, reserved bytes in FHS block are not zero, or options conflict with command.
03	Invalid logical unit. A Logical Unit Number (LUN) was specified which is larger than the maximum accommodated by the system. The maximum LUN is 8.
04	Volume error. Volume specified is not mounted.
05	Duplicate filename. On allocate call, file already exists.

TABLE 5-2. File Handling Services (FHS) Error Messages (cont'd)

zz	DESCRIPTION
06	File descriptor error. Filename, extension, or catalog not all alphanumeric with first character alphabetic.
07	Protect code error. Read/write protect codes in parameter block conflict with file/device codes.
08	Record length error. On an Allocate request, record length specified not even or too large for data block size.
09	Shared segment error. On an Assignment request, user's passed shared segment logical address conflicts with other address space or user has too many segments. Attempted Change-LUN request with shared segment.
0A	Insufficient directory space. On an attempt to allocate a file or rename a file, disk space is insufficient to insert the new directory entry.
0B	Access permission error. On an Assignment request, requested access permission conflicts with existing access permission. On other FHS requests requiring an assignment, file/device not assigned EREW.
0C	Insufficient system space. Not enough memory exists to allocate a segment for the file's data block and File Access Block (FAB). This error could occur on assignment if a shared segment is requested, or error could occur on first I/O request.
0D	Invalid assignment. On an Assignment request, LUN requested is already assigned. On other FHS requests requiring an assignment, LUN specified is not assigned.
0E	Invalid device type. An Allocate or Delete request was made for a non-random access device.
0F	Buffer overflow. On a Fetch-Device-Mnemonics request, the user parameter block specified for the returned information is not large enough to accommodate all device mnemonics.
10	Invalid task name. On a Change-LUN request, the task name or session specified in the parameter block is for a non-existent task.
11	Invalid buffer address. Address of user's parameter block for returned information on Fetch-Next-Directory-Entry or Fetch-Device-Mnemonics request is not in user's segment space or not in read/write segment, or address is odd.
12	Invalid file type. On an Allocate request, file type specified is invalid.

TABLE 5-2. File Handling Services (FHS) Error Messages (cont'd)

zz	DESCRIPTION
13	Internal FHS error caused by system problem.
14	Invalid parameter block address. User's FHS block not in his segment space or not in read/write segment, or address is odd.
15	Data block length error. On an Allocate request, the data block size specified for a non-contiguous file is less than 4.
16	Size error. On an Allocate request, size specified for a contiguous file is zero.
17	Non-existent filename. On an Assignment or Delete request, the file specified does not exist.
18	End of directory. Returned on a Fetch-Next-Directory-Entry request when no more directory entries exist.
19	Key length error. On an Allocate request, the key length specified for an indexed sequential file is less than 4 or greater than 100, not even, or greater than the record length.
1A	FAB length error. On an Allocate request, the FAB length specified for a non-contiguous file is greater than 20.
1B	Default volume not defined on Fetch-Default-Volume request.
1C	File not ready to output.
1D	User number not owner or user 0.

TABLE 5-3. Input/Output Services (IOS) Error Messages

zz	DESCRIPTION
00	No error.
81	IOS trap server does not exist. Most likely caused by system problem.
PARAMETER BLOCK ERRORS	
82	Invalid function. Function specified is not valid for device, reserved bytes in IOS block are not zero, or options conflict with function. Also occurs if current access permission does not permit requested function (i.e., write request with read only access permission).
83	Invalid logical unit. Logical unit specified is not assigned.
84	Invalid data buffer. Buffer starting address is odd or ending address is less than starting address. Can also occur if data buffer is incorrect size for specific I/O request (not multiple of 256 bytes for contiguous file or volume I/O, not equal to record size for fixed length records, larger than data block length for variable length records, larger than 256 bytes for record write with space compression, not equal to data block size for block read, or greater than data block size for block write). If a shared segment was requested at assignment, the error will occur if the data buffer specified on record I/O is in the same area as the shared segment or if the shared segment is not used when doing block I/O.
85	Invalid random record. On a block-I/O request to a noncontiguous file, the random record number specified on a random-I/O request does not correspond to the first sector of a data block. On a write request, no data is transferred. On a read request, the data block is transferred and the random record number in the IOS parameter block is updated to contain the logical sector number corresponding to the first sector in the data block that contained the requested sector.
86	Invalid parameter block address. User's IOS block not in his segment space or is not in read/write segment, or address is odd.
87	Protect code error. Write protected file targeted by write request. Volume read request made by non-owner of volume.

TABLE 5-3. Input/Output Services (IOS) Error Messages (cont'd)

zz	DESCRIPTION
DEVICE INDEPENDENT ERRORS	
C1	Buffer overflow. On a record-read request, user's buffer is not large enough to accommodate entire record, or for variable length record requiring space expansion, record is larger than 256 bytes.
C2	End of file. Attempted to read starting beyond end of file or attempted to write logical record or block starting beyond end of file plus one.
C3	End of volume. On device or volume I/O, attempted to start read beyond end of device; or on write request, attempted to write beyond end of device.
C4	Invalid or empty FAB. File integrity destroyed due to possible system failure.
C5	Invalid transfer for device. Device does not support option requested (e.g., binary or image request to device that does not support binary or image).
C6	Break condition. I/O was terminated by **BREAK** key.
C7	Internal I/O error caused by system problem.
C8	FAB/data block conflict. Data altered in shared segment specified at assignment. Possibly caused by system problem.
C9	Record does not exist. Logical record specified on record update or delete request does not exist. Record with key specified on random read request does not exist.
CA	Record already exists. Write record request specified existing record.
CB	Record overflow. Data altered in shared segment specified at assignment extending segment length beyond end of data block. Possibly caused by system problem.
CC	Key error or FAB/key conflict. Data altered in shared segment specified at assignment, causing keys to be out of sequence.
CD	Insufficient disk space. No space available on volume for FAB or data block or contiguous file space.
CE	Unrecoverable file error. Prior error caused file to be left in such a state that no more I/O can be done. File must be closed.
CF	File space allocation/deallocation conflict. FAB of file in error.

TABLE 5-3. Input/Output Services (IOS) Error Messages (cont'd)

zz	DESCRIPTION
DEVICE DEPENDENT ERRORS	
D1	Unrecoverable device error.
D2	Data compare error.
D3	Sector protect error.
E1	Device not ready.
E2	Device busy.
E3	Data Cyclic Redundancy Check (CRC) error.
E4	Write protected device. The diskette has the write protection tab removed.
E5	Deleted data mark detected.
E6	Time-out. Device did not respond within allotted time.
E7	Invalid sector address. Probably system problem. All invalid sector addresses normally detected before device driver is called.
E8	Checksum error. Data transfer error between MC68000 and an Intelligent Peripheral Controller (IPC).
E9	Disk restore error.
EA	Data overrun.
EB	Device status changed. Disk device that had been on-line went not ready and then ready again. Any files that had been open when this happened must be closed, without performing any more I/O. It is important to not open a disk drive door while any files are open because, if this happens, files can no longer be updated at close time. This may result in lost data as disk updates do not necessarily take place immediately, but may be kept in memory until close.
EC	Track/sector ID not found.
ED	Address mark CRC error.
EE	Seek error.
EF	Bad sector.

TABLE 5-3. Input/Output Services (IOS) Error Messages (cont'd)

zz	DESCRIPTION
CHANNEL ERRORS	
F1	Channel busy.
F2	Channel Direct Memory Access (DMA) error.
F3	Unrecoverable channel error.
F4	Controller error.
F5	Device configuration.
F6	DMA bus error.
F7	DMA mapping error.
F8	DMA controller error.

5

THIS PAGE INTENTIONALLY LEFT BLANK.

A

APPENDIX A

GLOSSARY OF TERMS

command pointer — The character >, which:

a. In CRT mode, resides in column 0 on line 22 of the screen. When the cursor is also on this line (for example, after pressing the **BREAK** or **F1** key), characters entered on the keyboard are interpreted as editor commands.

b. In line mode, is preceded by the letter E. The E> prompt is displayed at the left margin whenever the editor is awaiting command input.

CRT mode — The CRT editing mode has two levels:

Command level — is active when the cursor resides on line 22. Characters entered on the keyboard are interpreted as editor commands. Use of the **F1**, **F3**, **F4**, **F5**, **F6**, **F11**, **F12**, and all cursor control keys is also permitted.

Page level — is active when the cursor resides in the 20-record window. Characters entered on the keyboard are interpreted as text and placed in the edit file.

current line — A horizontal row of characters which:

a. In CRT mode, is where the record pointer resides.

b. In line mode, is the record displayed by an L * command. It is also the record displayed subsequent to various commands, when the verify function is enabled.

current record — Same as current line.

cursor — The blinking image on the monitor screen. It marks where a character typed on the keyboard is entered on the screen.

edit file — The body of logically sequential records upon which the editor can immediately act.

A

file — A body of records. It either has the same meaning as edit file, or it refers to a VERSAdos ASCII file.

line — A horizontal row of edit file data. It may be blank or contain text, and is terminated by an end-of-line character (usually, the RETURN character). Same as record.

line mode — The editing mode during chainfile execution or on non-EXORterm 155 or non-VME/10 terminals. It has two levels:

- Command level — is active when the prompt E> is displayed at the left margin. Characters input are interpreted as editor commands.
- Insert level — is active when the prompt > is displayed. Characters input are treated as text and inserted in the edit file. Insert mode cannot be used during chainfile execution.

line number — A logical record number given to each record in the edit file. The numbers start with 1 at the beginning-of-file, and increment by one to the end-of-file. They are not part of the file, but are used by the editor to communicate with the user.

record — Same as line.

record number — Same as line number.

record pointer — The character > in CRT mode, which resides in column 0 of the 20-record window. It is always on the line of the current record.

XTRACT buffer — A dynamic buffer, used for moving records within the edit file. Records are moved into the buffer, and then returned to the file by means of an **XTRACT** command.

When created by a **DUP** or **MOVE** command, its size is: (number of records) x 79 characters/record. For example, 10 records create a 790-character buffer.

When created by an **ADUP** or **AMOV** command, its size is 10 x (number of records) x 79 characters/record. For example, 10 records create a 7900-character buffer.

INDEX

INDEX

SUGGESTION/PROBLEM REPORT

MICROSYSTEMS

QUALITY • PEOPLE • PERFORMANCE

Motorola welcomes your comments on its products and publications. Please use this form.

To: Motorola Inc.
Microsystems
2900 S. Diablo Way
Tempe, Arizona 85282
Attention: Publications Manager
Maildrop DW164

Product: _____ Manual: _____

COMMENTS: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Please Print

Name _____ Title _____

Company _____ Division _____

Street _____ Mail Drop _____ Phone _____

City _____ State _____ Zip _____

For Additional Motorola Publications
Literature Distribution Center
616 West 24th Street
Tempe, AZ 85282
(602) 994-6561

Four Phase/Motorola Customer Support, Tempe Operations
(800) 528-1908
(602) 438-3100

MOTOROLA

MOTOROLA ***Semiconductor Products Inc.***

P.O. BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

19142 PRINTED IN USA (1/86) MESSENGER 3500